

Requisitos: domine-os!

ANDRADE JR., Ademir
02/09/2016

“Requisitos são o ‘o que’. Design é o ‘como’”
- Autor desconhecido

Vários são os fatores de fracasso de um projeto, mas os principais sempre serão o arrasto de escopo, má comunicação, falta de comprometimento dos *stakeholders* e apoio inadequado do Patrocinador do Projeto. Todos esses fatores têm algo em comum: eles envolvem, direta ou indiretamente, requisitos.

Um exemplo de fracasso de projeto por problema de requisitos é o foguete Ariane 5. SANKAR (2013) e DONALD (2000) relatam que em 1996, a estréia da nova versão dos foguetes da série Ariane, da Agência Espacial Européia (ESA), tornou-se um desastre por conta de uma conversão de software mal feita. 37 segundos após o lançamento o foguete entrou em auto-destruição e causou uma perda de 500 milhões de dólares. O projeto todo custou 7 bilhões de dólares.

Em outros casos, o cuidado com os requisitos pode levar o projeto ao sucesso: SANKAR (2013) nos conta que a fabricante de aeronaves Boeing se aliou a oito companhias aéreas para definir e modelar os requisitos de projeto para a fabricação de um novo avião, o Boeing 777, que se tornou um sucesso comercial na história da empresa.

Mas, afinal, o que é um requisito?

Requisitos são condições e capacidades físicas ou funcionais necessárias para resolver um problema, atingir um objetivo, cumprir um contrato, satisfazer uma especificação, entre outros. Requisitos regem grande parte do projeto – o escopo, os custos, o tempo e a qualidade do projeto orbitam os requisitos.

SANKAR (2013) cita cinco grandes ameaças comuns a projetos:

- a) Falta de entendimento dos requisitos;
- b) Falta de comunicação contínua entre os *stakeholders*;
- c) Falta de consenso na definição dos requisitos;
- d) Falta de envolvimento de outras áreas;
- e) Arrasto de requisitos (falta de limitação de escopo).

O mesmo autor ainda completa, citando características que um conjunto de requisitos deve possuir:

- a) Estabelecer um consenso entre o patrocinador do projeto, clientes e outros *stakeholders*;
- b) Aumentar a confiança do cliente no produto a ser entregue;
- c) Prover um *roadmap* para o desenvolvimento do projeto.

No entanto, requisitos não são um “pacote fechado”, o qual você monta uma vez e não tem mais a chance de revisá-lo. Projetos mudam a todo tempo, principalmente porque **seus requisitos mudam frequentemente**. Essas mudanças não só são aceitáveis como há métodos de gerenciá-las, como será visto posteriormente.

A quinta grande área do PmBOK, Gerenciamento de Escopo, traz como um dos processos “Coletar os Requisitos” (5.2). Segundo o próprio, “O principal benefício deste processo é o fornecimento da base para definição e gerenciamento do escopo do projeto, incluindo o escopo do produto”. Com isso, vemos que este processo é de suma importância para o projeto pois a sua saída, ou produto, serve como base tanto para a Declaração de Escopo do Projeto como também para limitar o **escopo do próprio produto**.

A figura 1 ilustra as entradas, técnicas e saídas deste processo.

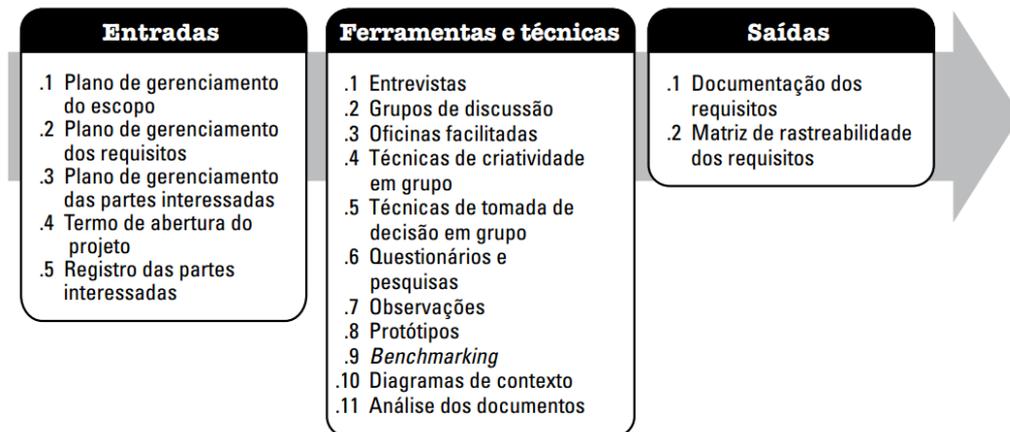


Figura 1 - Coletar os requisitos: entradas, ferramentas e técnicas, e saídas

Como descrito na figura acima, várias são as ferramentas e técnicas utilizadas para estabelecer os requisitos do projeto. A grande maioria delas envolve efetivamente **comunicação entre as partes interessadas**. O objetivo destas técnicas é justamente fazer com que todos os *stakeholders* sejam envolvidos e que os requisitos, além de compreendidos por todos, sejam principalmente aceitos em consenso.

Requisitos são classificados durante o levantamento dos mesmos. A classificação depende muito do tipo de negócio, de projeto, de modelo sendo utilizado. Abaixo, segue um exemplo citado pelo BABOK (Business Analysis Book Of Knowledge, 2009):

- *Requisitos de Arquitetura:* explicam o que deve ser feito identificando a integração necessária entre a **estrutura** e o **comportamento** do sistema, ou seja, a **organização fundamental de um sistema**;
- *Requisitos de Negócio:* são descrições dos objetivos, justificativas e necessidades de uma organização. Normalmente, estes requisitos descrevem os problemas que a organização quer resolver ou as oportunidades que ela enxerga. Estas informações podem ser obtidas também em outros documentos, como o de **caso de negócio**.
- *Requisitos de Usuário:* é a declaração de um *stakeholder* ou grupo de *stakeholders* sobre suas necessidades. É um tipo de requisito que serve como uma “ponte” entre um

requisito de negócio e um requisito **funcional**, em casos em que a necessidade de uma pessoa ou grupo de pessoas é fundamental no processo.

- **Requisitos Funcionais:** descrições detalhadas do **comportamento** e informação que a solução precisará. Exemplos de requisitos funcionais incluem formatação de texto, cálculo de valores, seleção de imagens.
- **Requisitos Não-funcionais:** descrições detalhadas das **condições** sob as quais a solução deve ser eficaz, **qualidades** que a solução deve ter, ou **restrições** sob as quais a solução deve operar. Exemplos de requisitos não-funcionais são confiabilidade, robustez, segurança, disponibilidade.
- **Requisitos de Implementação:** descrições sobre comportamentos e situações necessárias **apenas no momento da transição do estado atual da empresa para o estado após o projeto**. Por exemplo, recrutamentos, qualificações, mudanças de setor, migração de dados entre servidores são requisitos de implementação, chamados também de Requisitos de Transição.

Indiferente de como os requisitos são classificados, eles devem ter um conjunto de características que os definem como "bons requisitos". Vários autores discorrem sobre estas características, mas as mais comuns são as seguintes.

- a) **Unitário** ou **Coeso**: o requisito se refere a somente um assunto.
- b) **Completo**: o requisito é totalmente exposto em um lugar, sem falta de informação.
- c) **Consistente**: o requisito não pode contradizer nenhum outro, ou nenhuma outra documentação externa.
- d) **Disjunto**: o requisito não pode conter conjunções, ou ligações. Por exemplo, um requisito do tipo "O campo CNPJ deve validar o código brasileiro e o uruguaio (RUT)". Neste caso, deve-se separar o requisito em "(1) O campo CNPJ deve validar o código brasileiro" e "(2) O campo CNPJ deve validar o código uruguaio (RUT)".
- e) **Rastreável**: o requisito atende às necessidades de negócio conforme declarado pelos *stakeholders* e foi documentado por escrito.
- f) **Atual**: o requisito não ficou obsoleto com o passar do tempo do projeto.
- g) **Sem ambiguidade**: o requisito é escrito em linguagem concisa, sem recorrer a jargões técnicos, siglas, ou linguagem rebuscada. Expressa fatos, não opiniões. Expressões negativas (do tipo "a funcionalidade X não pode ser...") devem ser evitadas. Neste ponto é importante expressar a máxima da administração: o que é óbvio para uma pessoa pode não ser para outra.
- h) **Importância específica**: o requisito deve ter claramente especificado se a presença ou ausência dele é crucial para o andamento do projeto ou produto final, indicando mesmo uma deficiência fatal, ou se ele pode ser executado se o cronograma ou o orçamento permitirem.
- i) **Verificável**: a implementação do requisito pode ser verificada por métodos básicos de medição, análise e testes.

Um dos maiores problemas no levantamento de requisitos é saber separar *requisitos de design*. Requisitos, como já dito, são um conjunto de condições para resolver um problema. No

Design é especificada a implementação de soluções que satisfaçam os requisitos. Como a citação que abre este artigo nos traz, o documento de requisitos descreve **o que** fazer, enquanto o documento de design descreve **como** fazer o que foi descrito no documento de requisitos.

Um dos papéis mais importantes do Documento de Requisitos é prover uma **linha de base** para o projeto. A linha de base dá forma à Declaração de Escopo e, assim, podemos monitorar **mudanças no projeto**.

Diz-se que, em projetos, mudanças são uma constante. Isso de fato é uma variável a ser levada em consideração desde o início do planejamento. Existem dois tipos principais de mudanças: as **externas** ao projeto/produto (mudanças no mercado, legislação, ambiente político/econômico, recursos da organização, por exemplo), e as **internas** do projeto. Essas mudanças podem acontecer por alguns fatores não previstos pelos *stakeholders*. Um dos trabalhos mais importantes de um gerente de projetos, no gerenciamento de requisitos, é manter a comunicação clara e o pleno entendimento das partes envolvidas. Para tanto, existe o TAP, o Caso de Negócio, a Declaração de Trabalho, entre outros documentos, que darão sustentação a esse entendimento. Assim, evita-se o excesso de mudanças no projeto e por consequência alterações em cronograma, orçamento, etc.

Mas, com todo o cuidado que possamos ter, as mudanças virão e temos de controlá-las. Como fazê-lo?

O processo de **controle de mudanças** precisa ser formal. Os passos para o controle podem variar de empresa para empresa, mas grosso modo envolvem (ROUSE, 2011):

- a) Documentação do pedido de mudança: quando um *stakeholder* requisita a mudança, é categorizada e registrada, junto com uma avaliação breve da mudança, sua importância, e a dificuldade de implementá-la.
- b) Avaliação formal: a justificativa para a mudança e os riscos e benefícios de realizar ou não a mudança são avaliados. Se a mudança for aprovada, uma equipe de desenvolvimento é designada. Se for rejeitada, o fato é documentado e o requisitante, comunicado.
- c) Planejamento: a equipe responsável pela mudança cria um plano detalhado de implementação, bem como um **plano para desfazer a mudança, caso ela se torne prejudicial ao projeto**.
- d) Implementação e testes: a equipe responsável implementa a mudança. Se a equipe de testes aprovar, libera para implantação no cliente.
- e) Implantação e revisão: a equipe responsável realiza a implantação e os *stakeholders* revisam-na.
- f) Avaliação Final: se o requisitante ficar satisfeito com a mudança, o pedido de mudança é encerrado. Se não, o projeto é reavaliado e alguns passos podem ser repetidos.

Um dos principais problemas nas mudanças de requisitos é o **arrasto de requisitos**, ou **arrasto de escopo**. Isso acontece principalmente quando:

- a) Os *stakeholders* não conhecem bem os objetivos do projeto;
- b) O cliente cria uma "lista de desejos" que nunca para de crescer;
- c) Os desenvolvedores enxergam no projeto uma oportunidade para aperfeiçoamento do produto.

Mesmo com o controle de mudanças bem estruturado, esses fatores contribuem para um “projeto sem data para acabar”. Muitas vezes é necessário até mesmo encerrar a absorção de mudanças para o bem do andamento do projeto. Lembre-se que arrasto de escopo muitas vezes significa **arrasto de cronograma** e **arrasto de custos**.

Nosso caso de uso: Projeto Verão

Em um artigo anterior, falamos sobre a construção da EAP do Projeto Verão. Lá, conversamos sobre algumas coisas que o projeto teria que ter para chegarmos ao resultado desejado, como frequentar uma academia, melhorar a alimentação, ter um meio de transporte que não fosse um carro. Para criarmos um Documento de Requisitos deste Projeto, faremos um documento com um layout semelhante ao abaixo:

Projeto Verão	
Documento de Requisitos	
Criador: Regiane	Data: 09/09/2016

ID	Classificação	Descrição	Prioridade
1	ACADEMIA	A academia deve ser na cidade de São Ludgero.	Necessário
2	ACADEMIA	A mensalidade deve ser, no máximo, R\$ 150,00/mês.	Necessário
3	ACADEMIA	A academia pode ser exclusiva para mulheres.	Desejável
4	ALIMENTAÇÃO	Uma nutricionista deve ser consultada antes de iniciar dieta.	Necessário
5	ALIMENTAÇÃO	A dieta não pode conter glúten (restrição alimentar). (1)	Necessário
5	ALIMENTAÇÃO	A dieta deve seguir restrição de glúten.	Necessário
6	TRANSPORTE	O meio de transporte deve ser sustentável. (2)	Necessário
6	TRANSPORTE	O meio de transporte deve ser também um tipo de atividade física.	Necessário

No exemplo acima, a tabela de requisitos possui um **ID** e uma **Classificação**. Estes campos são utilizados para manter a **rastreabilidade** dos requisitos. No campo **Descrição**, preenche-se o requisito em si. A **Prioridade** serve para ordenar, por importância, os requisitos. Esta nomenclatura varia de organização para organização; o importante é definir o que é indispensável para o projeto – o requisito sem o qual o projeto não tem razão de ser – e o que pode ser realizado dependendo das restrições de tempo e custos, sem prejudicar a qualidade do projeto.

Foram dados dois exemplos de requisitos “ruins”. O requisito 5 foi escrito de uma forma negativa (1). Logo abaixo o requisito foi reescrito de forma positiva, de forma a evitar que “tudo o que não pode, pode”. Já no requisito 6, houve uma ambiguidade (2). Sustentabilidade pode ser alcançada de várias formas, inclusive algumas que não contribuem para a saúde física. O requisito foi reescrito de forma a garantir que o meio de transporte utilizado seja também um exercício.

Como visto, o bom gerenciamento de requisitos são um dos fatores que garantem o sucesso de um projeto. Esta área requer um alto nível de comunicação e entendimento entre todas as partes interessadas do projeto. Aqui, "prevenir é melhor que remediar" é a tônica.

REFERÊNCIAS BIBLIOGRÁFICAS

ARNOLD, Douglas. *The Explosion of Ariane 5*. Disponível em <<https://www.ima.umn.edu/~arnold/disasters/ariane.html>>. Acesso em 23/08/2016.

IIBA. *A Guide to the Business Analysis Body of Knowledge® (BABOK® Guide) Version 2.0*. 2009.

PMI. *Um Guia do Conhecimento em Gerenciamento de Projetos (Guia PMBOK®)*. 5ª Ed. – EUA: Project Management Institute, 2013.

ROUSE, Margaret. *Change Control*. Publicado em Janeiro de 2011. Disponível em <<http://searchdisasterrecovery.techtarget.com/definition/change-control>>. Acesso em 09/09/2016.

SANKAR, Vijay. *Managing Your Requirements 101 – A Refresher Part 1: What is requirements management and why is it important?* Disponível em <<https://www.ibm.com/developerworks/community/blogs/requirementsmanagement/entry/why-requirements-management?lang=en>>. Acesso em 23/08/2016.

SMITH, Aaron. *Requirements Management: A Core Competency for Project and Program Success*. EUA: Project Management Institute, 2014.